Dynamic Trust Management Framework for Robotic Multi-Agent Systems

Igor Zikratov[1], Oleg Maslennikov[1], Ilya Lebedev[1], Aleksandr Ometov[2(✉)], and Sergey Andreev[2]

[1] Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (ITMO University), St. Petersburg, Russia
[2] Tampere University of Technology, Korkeakoulunkatu 10, 33720 Tampere, Finland
aleksandr.ometov@tut.fi

Abstract. A lot of research attention has recently been dedicated to multi-agent systems, such as autonomous robots that demonstrate proactive and dynamic problem-solving behavior. Over the recent decades, there has been enormous development in various agent technologies, which enabled efficient provisioning of useful and convenient services across a multitude of fields. In many of these services, it is required that information security is guaranteed reliably. Unless there are certain guarantees, such services might observe significant deployment issues. In this paper, a novel trust management framework for multi-agent systems is developed that focuses on access control and node reputation management. It is further analyzed by utilizing a compromised device attack, which proves its suitability for practical utilization.

Keywords: Multi-agent systems · Information security · Access control · Trust management

1 Introduction

Today, swarm multi-agent robotics is one of the most significant and complex fields of research, especially in light of the fact that only under 5 % of our planet, both land and oceanic, has been explored so far[1]. Modern robots employed for surface research, protection, and monitoring are extremely complicated devices equipped with a variety of sensing mechanisms [1]. Therefore, it is important to keep them operational autonomously for as long as possible[2].

For example, wildfire fighting remains one of the most physically challenging tasks faced by human-workers today. Autonomous machines can contribute significantly to facilitate this hard, dirty, exhausting, and dangerous job [2]. Robotic devices can often operate faster and more efficiently while keeping people away

[1] See: NOAA National Ocean Service: How much of the ocean have we explored? 2014. http://oceanservice.noaa.gov/facts/exploration.html.
[2] See: Autonomous Fire Guard (AFG) concept. 2009. http://www.yankodesign.com/2009/08/21/firefighters-best-friend/.

O. Galinina et al. (Eds.): NEW2AN/ruSMART 2016, LNCS 9870, pp. 339–348, 2016.
DOI: 10.1007/978-3-319-46301-8_28

from unsafe locations[3]. Conventionally, such devices are expected to cooperate with each other in order to reach common "targets" in remote locations [3].

Such distributed coordination has many advantages in reaching common group goals, lower operating costs, control system requirements, improve robustness, and achieve better scalability [4,5]. The related aspects have been widely recognized and well-studied over past years [6,7]. In multi-agent systems, the actual network topology connecting all the devices in operation plays a crucial role in determining consensus. Typically, the objective is to explicitly identify necessary and sufficient conditions for a particular network topology, such that a common agreement could be reached under specifically designed algorithms.

One of the most promising trends in modern robotics is the development of management tools that allow for intelligent Multi-Agent System (MAS) group control. In particular, considerable research attention has been paid to the collaborative planning frameworks, which operate in decentralized, "ad hoc" regime by forming a coalition [8]. This is to achieve better scalability, operational coverage, and network availability in cases of weak connectivity to the control unit. A significant body of research literature in this field has been dedicated to dynamic redistribution of goals between the operating nodes in case of their unpredictable breakdowns [9].

Due to the "ad hoc" nature of the considered networks, which often operate in a dynamic mesh fashion, the MAS becomes an attractive field for a wide range of attacks, such as: message capture and retransmission, violation of integrity, unauthorized data access, denial of service, etc. [10,11]. Hence, the currently utilized trust management schemes may be very limited due to discretionary distinction and mandate-based behavior [12,13]. We subdivide the possible attacks on a MAS into the following main categories [14]: (i) network-layer related attacks; (ii) attacks on identification and authentication of agents in the system [15]; and (iii) *compromised device* intrusion [16]. The main goal of this work is thus to develop a trust management framework suitable for resisting the harmful compromised device attack.

This paper is organized as follows. In Sect. 2, we review the decentralized MASs and the corresponding attacks. Further, in Sect. 3, we introduce a trust model resistant to the discussed type of attacks. Then, in Sect. 4, the compromised device intrusion attack detection in detailed. The last section describes our future work and offers some conclusions.

2 Modeling Background

In this section, we consider a MAS operating in a decentralized fashion [17,18]. We focus on a group of N robots targeting a common collaborative goal. During the initialization phase, each of the devices receives its utility function (goal) related data. The overall framework operation model is captured in Fig. 1.

[3] See: The National Interagency Fire Center (NIFC): Incident Management Situation Report. 2016. http://www.nifc.gov/nicc/sitreprt.pdf.

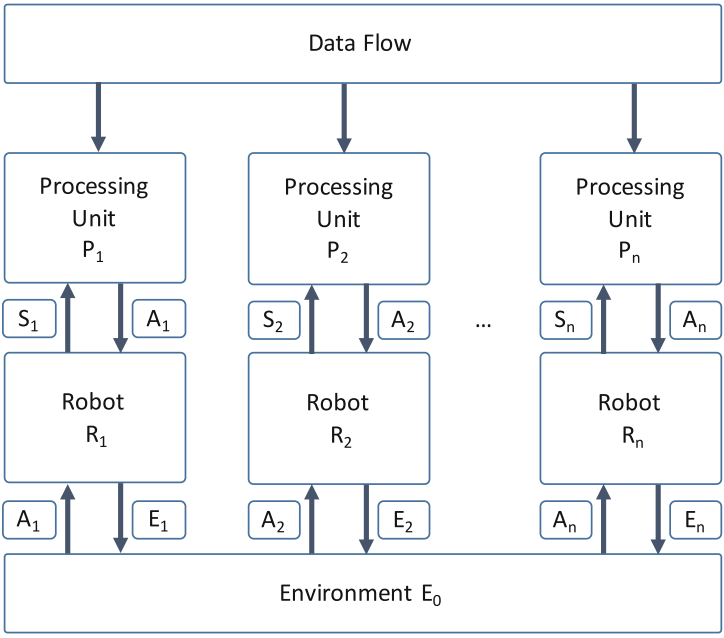

Fig. 1. Simplified decentralized MAS management framework.

2.1 Preliminary Assumptions and Definitions

Each processing unit $P_i, (i = \overline{1, N})$ of every device R_i consists of the corresponding computing unit CU_i, the data transmitting unit DT_i, the data receiving unit DR_i, the current state determination unit CS_i, and a set of sensing modules SD_i. The CU_i is communicating with other CU_j by transmitting the system state information S_i^0 and the respective operating decisions $A_i^{k+1}, (k = 0, 1, 2, \ldots, N)$. In addition, each CU_i has the knowledge of the environmental data E_i^0 and its own state S_i^0 to continuously update the utility function ΔY for any possible operating decisions in the current state. We select $max(\Delta Y)$ to be our utility function.

As an attack, we consider any malicious activity of the compromised device on the k^{th} iteration of the system operation [19]. As a result, the following device decision A_i^{k+1} might not be selected according to the utility function. We also consider the attack with message capture and retransmission, as well as the attack on the environment estimation and the attacks targeted to affect the group decision making protocols [20].

Conventionally, a MAS utilizes the following techniques to enable secure "ad hoc" communication: a state-estimation function [21]; the lightweight cryptography solutions [22]; the time-limiting techniques [23]; and the Buddy Security Model (BSM) [24,25], among others. Interestingly, the neighboring nodes in e.g., BSM are responsible for each other's security by monitoring their environment

continuously. This is reached by means of exchanging specialized *tokens* between the BSM users that contain confidential state information, as well as monitoring any potential security threats from the surrounding devices. By informing the neighboring nodes about anomalous behavior of a new device, each agent contributes its share of stability to the overall system security.

Today, the BSM is receiving increased attention primarily due to its decentralized nature. On the other hand, utilization of this model in the robotic systems could still be affected by a compromised robot. Our scenario of interest relates to the "ad hoc" network operation in remote areas, where providing reliable connection to the centralized control unit may be a challenging task. Therefore, physical capture of a device and compromising a token become possible. In this work, we develop an improved BSM formulation by introducing a device's trust level that makes it more difficult to perform the known attacks [26].

2.2 Multi-Agent Trust Model for Robotic Systems

In what follows, we describe the MAS operation in a steady-state regime i.e., past the initialization phase. In the current state S_i^0, each i^{th} robot $R_i, (i = \overline{1,N})$ collects the data from $CU_j, (i \neq j, j = \overline{1,N})$ of other robots in the group. After this phase, the robot in question selects A_j^{k+1} according to the utility function ΔY and reports the corresponding decision to other devices with $w(A_i^{k+1})$ to $CU_j, (i \neq j, j = \overline{1,N})$. This message is based on the received information $S_1^0, S_2^0, \ldots, S_i^0 - 1, S_i^0 + 1, \ldots, S_N^0$ and the possible current decisions $A_1^{k+1}, A_2^{k+1}, \ldots, A_i^{k+1} - 1, A_i^{k+1} + 1, \ldots, A_N^{k+1}$. After receiving the message, other agents are validating the data with respect to the decision made by the i^{th} node.

In case the above check by the j^{th} device resulted in $\Delta Y_j, (i \neq j) \neq \Delta Y_i$, the trust level of the i^{th} device is increased. We define the trust level as *willingness* of a particular node to report valid information to others. Alternatively, if a device has reported a faulty ΔY_i, its level of trust is decreased. As a result, we may now define a new parameter, which is a set of "steps" l required to estimate the *long-term* level of trust per device A_i^{l+1}. Therefore, when calculating ΔY_i^l at the following system iteration, the devices would rely more on nodes with higher trust levels.

In summary, the lower levels of trust would not allow the compromised robots affect the system operation in a destructive way even when sending a valid-like token. Thereby, potential malicious nodes will have to behave similar to the trusted ones for the interval of time that is necessary to build sufficient trust, which improves system robustness to the considered type of attacks.

3 Trust Model Development

In this section, we first discuss the notation related to our security mechanism formulation and then outline its implementation possibilities. Our developed trust model is illustrated in Fig. 2, where arrows represent multi-agent connections over alternative channels, such as visual, NFC, etc. Further, the wireless radio links are indicated with the dashed lines.

To ease the exposition, we introduce the notation used in this section as follows: $A = \{A_1, A_2, \ldots, A_N\}$ are the operations that may be performed by an agent; $S = \{s_1, s_2, \ldots, s_N\}$ is the set of states during the communication phase; $V = \{F, T\}$ is the set of results given by the report validation, where F corresponds to a faulty reply and T stands for a valid one; and r_l^m is the trust level determined by the m^{th} agent for the l^{th} device ($l \neq m$). There are different options to implement this model according to Fig. 2.

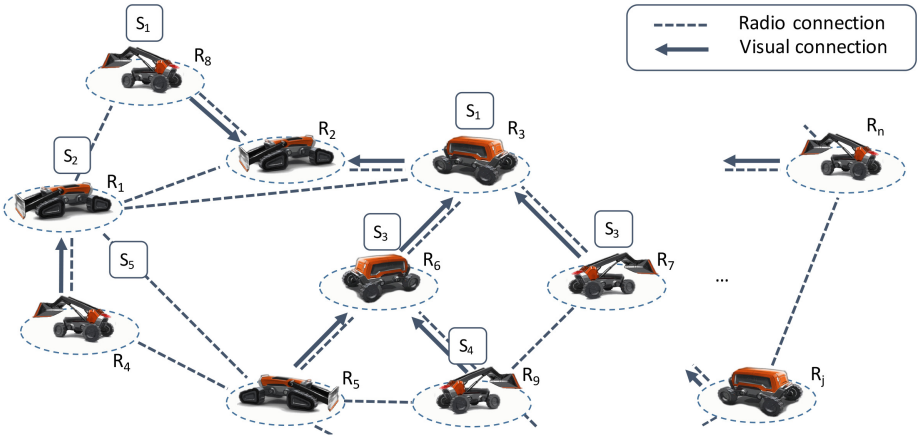

Fig. 2. Proposed trust model for interacting agents.

As an example, we assume that the system is in the state k and focus on the $2nd$ device in the network. It has reported the message $2A_i$ to the rest of the users. As shown in the figure, the $2nd$ robot has a wireless connection to the devices 1, 3, and 8, and all of these users have received the corresponding message. Further, the devices 3 and 8 have a visual proof that the $2nd$ device has indeed performed the reported action. The set of possible system states S could then be represented as follows:

- s_1: the object is in the line-of-sight conditions and can be reached via a radio link (for devices 3 and 8);
- s_2: the object is not in the line-of-sight conditions and can be reached via a radio link (for device 1).

If the $2nd$ device is acting normally, its actual operation A_2 is the same as the reported one i.e., the devices 3 and 8 should increase their levels of trust for the $2nd$ robot by $\Delta r_2^3 = \Delta r_2^8$ due to having a visual confirmation of the action. The device 1 has also received the report from the $2nd$ device, but it is only utilizing a wireless channel. In this case, the corresponding level of trust should be updated less significantly than for other nodes $\Delta r_2^1 < \Delta r_2^{3,8}$. Here, only the devices having a direct connection to the robot in question are evaluating its level of trust, but for the rest it remains unchanged.

Second, the devices having a direct connection to the robot in question may forward their knowledge to the neighbors. They would then transmit a message of type $i : A_i s_k^j v$, where i is the identifier of a reporting device, A_i is the value reported by the i^{th} device, s_k^j is the j^{th} device's state, and v is the validation result (either *true* or *false*). For our example, the corresponding messages should be: from the robot 1, $2 : A_i s_2^1 T$; from the robot 3, $2 : A_i s_1^3 T$; from the robot 8, $2 : A_i s_1^8 T$. These messages are delivered to the agents $4, 5, 6, 7, 9$. Note that for different "ad hoc" topologies and depending on the network dynamics, the results of such message distribution may vary. Hence, the set of states S for this case could be represented as:

- s_1: the object is in the line-of-sight conditions and can be reached via a radio link (for devices 3 and 8);
- s_2: the object is not in the line-of-sight conditions and can be reached via a radio link (for device 1);
- s_3: the subject s_1 is in the line-of-sight conditions and can be reached via a radio link (for device 6 and 7);
- s_4: the subject s_1 is not in the line-of-sight conditions and can be reached via a radio link (for device 9);
- s_5: the subject s_2 is in the line-of-sight conditions and can be reached via a radio link (for device 1);
- s_6: the subject s_2 is not in the line-of-sight conditions and can be reached via a radio link (for device 1).

The subjects with different states $s_1, s_2, \ldots, s_6$ would obtain different trust levels towards the same subject based on the possibility to evaluate the device by themselves. An increment-based trust scale for the i^{th} object may then be introduced as:

$$\Delta r_{s_1}^i > \Delta r_{s_2}^i > \cdots > \Delta r_{s_6}^i. \tag{1}$$

The main target of the indirectly connected agents is to determine their own state and select the objective level of trust based on it. Correspondingly, if the value of $v = T$, the level of trust is increased by $\Delta r_{s_i}^n$. Similarly, if $v = F$, the level of trust is decreased by the same value. Importantly, the reception of faulty data from different nodes may be caused by a variety of factors, including uncontrollable interference, severe weather conditions, and deliberate distortion of the message by one of the relaying devices. In the latter case, the collective goal may be achieved by utilizing the discussed MAS information security framework.

In summary, the second implementation option for the proposed trust model should be more efficient in practice. It allows delivering the actual trust information to a higher number of agents, but at the same time may lead to the increased amounts of signaling.

4 Detecting Attacks on a MAS

The trust level management mechanism proposed in the previous section allows controlling such threats as: capture, modification, and retransmission of messages

A_i and S_i, as well as compromising the system operation by an attacker node, which may attempt to influence the utility function ΔY. On the other hand, some MAS management protocols enable to select a *leading* device from within the group, which becomes responsible for handling system-wide decision-making functionality [27,28]. To achieve this, said device updates the utility function for the entire network.

As in any deployment with a single-node failure possibility, should an attacker seize this role, it may disrupt the operation of the entire system. Such an attack may also be executed by a set of devices within the group. In this case, the only way to detect the malicious behavior is by monitoring ΔY by all the system agents, and at each iteration of the network operation. In order to perform this monitoring, all the agents (except for the reporting one) need to recalculate the potential ΔY_{t+1} of the i^{th} node at the l^{th} step. In case $\Delta Y_{t+1} < \Delta Y_t$, the receiving device decreases the level of trust for the reporting robot by $\Delta r^i_{Y_l}$ and vice versa. The resulting level of trust for the i^{th} device could be calculated as:

$$\Delta\mu^i > \alpha\Delta r^i_{s_i} + \beta\Delta r^i_{Y_l}, \tag{2}$$

where α and β are the weights corresponding to the reported agent, and the utility of its decision is selected according to the information security policy of the MAS.

5 Selected Numerical Results

In order to validate the usability of our proposed model, we conduct a set of simulation runs utilizing the V-REP robotics framework[4]. To prove the effectiveness, we compare the changes of ΔY^l_i throughout the framework operation according to Sect. 3.

The initial system setup is described as follows. Each agent has a complete knowledge of the MAS goals, the corresponding distances between the agents, and the number required to reach the goal per each target. After all the agents have exchanged the relevant data, each of them is selecting the closest target by comparing its R and the corresponding other agents' R_{min} distances to it. If $\Delta Y^l_i = A_i(min) - A_i$ is positive, the agent in question reports on its decision to proceed with the current target; otherwise, it waits.

For the sake of our experiment, we employ 50 agents uniformly distributed across the circular area with the radius of 50 m. The number of targets is three, with 5, 3, and 2 required agents, respectively. Each agent has the radio coverage of 30 m and the line-of-sight distance of 7 m. We vary the number of compromised nodes in the system to validate our framework operation.

Regardless of the system type, the utilization of the proposed trust model reduces the impact of attacks on the system efficiency (see Figs. 3 and 4). For the second option, where each agent has at least one neighboring node with the visual contact, the benefits are even more significant (Fig. 4). The developed

[4] See: V-REP http://www.k-team.com/mobile-robotics-products/v-rep.

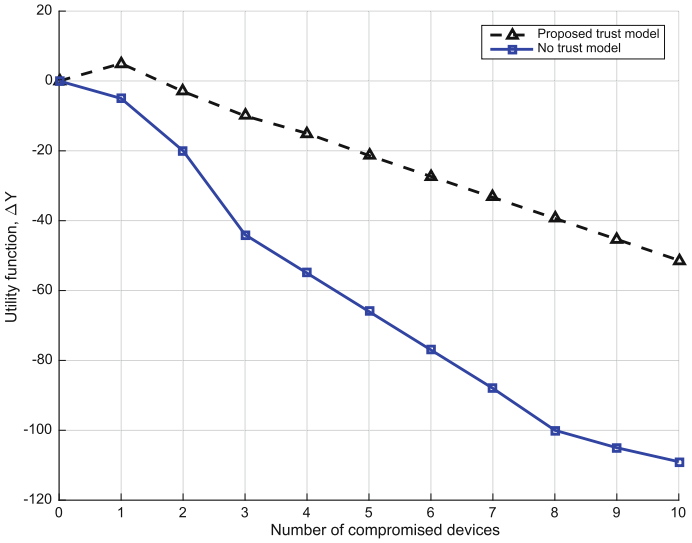

Fig. 3. Dependence of utility function on the number of compromised devices (uniform distribution of agents).

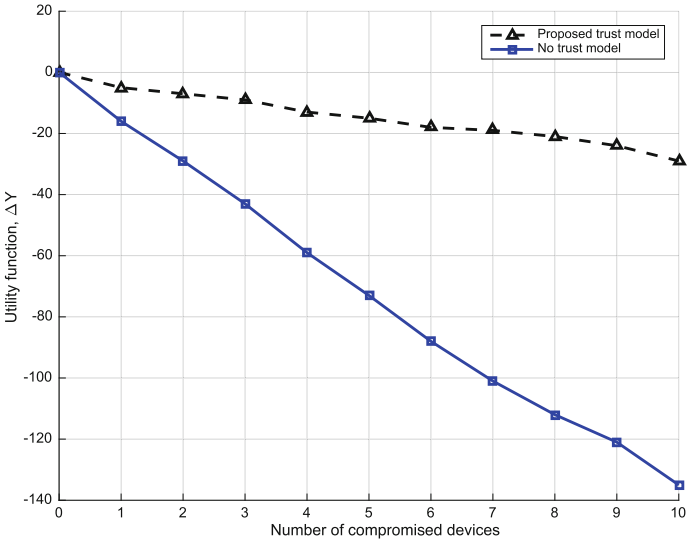

Fig. 4. Dependence of utility function on the number of compromised devices (each agent has at least one visual neighbor).

model may, however, have a negative impact on the system efficiency e.g., when the compromised node has been the best possible selection for the target.

6 Conclusions

In this work, we developed a trust model for the decentralized robotic MAS networks. Our framework provides group access based on the device-centric level of trust, which is selected and dynamically updated over time. Our formulation was successfully evaluated with simulations and may be utilized in modern MAS deployments. The main advantage of the proposed approach is in that it allows for continuous and secure communication in the robotic "ad hoc" networks that face the lack of reliable connectivity to the centralized control unit. The second benefit is in the time-driven dynamic trust level updates that determine the trust level actuality i.e., a newly-joined device would have to operate trustfully for a considerably long time in order to achieve any significant decision-making privileges.

References

1. Hernandez, L., Baladron, C., Aguiar, J.M., Carro, B., Sanchez-Esguevillas, A.J., Lloret, J., Chinarro, D., Gomez-Sanz, J.J., Cook, D.: A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants. IEEE Commun. Mag. **51**(1), 106–113 (2013)
2. Andreev, S., Larmo, A., Gerasimenko, M., Petrov, V., Galinina, O., Tirronen, T., Torsner, J., Koucheryavy, Y.: Efficient small data access for machine-type communications in LTE. In: Proceedings of the IEEE International Conference on Communications (ICC), pp. 3569–3574. IEEE (2013)
3. Cao, Y., Yu, W., Ren, W., Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination. IEEE Trans. Ind. Inf. **9**(1), 427–438 (2013)
4. Militano, L., Fitzek, F., Iera, A., Molinaro, A.: On the beneficial effects of cooperative wireless peer-to-peer networking. In: Pupolin, S. (ed.) Wireless Communications 2007 CNIT Thyrrenian Symposium, pp. 97–109. Springer, Heidelberg (2007)
5. Petrov, V., Andreev, S., Turlikov, A., Koucheryavy, Y.: On IEEE 802.16m overload control for smart grid deployments. In: Andreev, S., Balandin, S., Koucheryavy, Y. (eds.) NEW2AN/ruSMART 2012. LNCS, vol. 7469, pp. 86–94. Springer, Heidelberg (2012)
6. Ren, W., Beard, R.W., Atkins, E.M.: A survey of consensus problems in multi-agent coordination. In: Proceedings of the 2005 American Control Conference, pp. 1859–1864. IEEE (2005)
7. Lesser, V.R.: Reflections on the nature of multi-agent coordination and its implications for an agent architecture. Auton. Agent. Multi-Agent Syst. **1**(1), 89–111 (1998)
8. Shehory, O.M., Sycara, K., Jha, S.: Multi-agent coordination through coalition formation. In: Singh, M.P., Rao, A., Wooldridge, M.J. (eds.) ATAL 1997. LNCS, vol. 1365, pp. 143–154. Springer, Heidelberg (1998)
9. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
10. Jung, Y., Kim, M., Masoumzadeh, A., Joshi, J.B.: A survey of security issue in multi-agent systems. Artif. Intell. Rev. **37**(3), 239–260 (2012)
11. Araniti, G., Calabro, F., Iera, A., Molinaro, A., Pulitano, S.: Differentiated services QoS issues in next generation radio access network: a new management policy for expedited forwarding per-hop behaviour. In: Proceedings of the IEEE 60th Vehicular Technology Conference (VTC2004-Fall), vol. 4, pp. 2693–2697. IEEE (2004)

12. Bell, D., LaPadula, L.: Secure Computer Systems: Unified Exposition and Multics Interpretation, vol. MTR-2997 R. MITRE Corp., Bedford (1976)
13. Harrison, M.A., Ruzzo, W.L., Ullman, J.D.: Protection in operating systems. Commun. ACM **19**(8), 461–471 (1976)
14. Higgins, F., Tomlinson, A., Martin, K.M.: Threats to the swarm: security considerations for swarm robotics. Int. J. Adv. Secur. **2**(2&3), 1–10 (2009)
15. Petrov, V., Edelev, S., Komar, M., Koucheryavy, Y.: Towards the era of wireless keys: how the IoT can change authentication paradigm. In: Proceedings of the IEEE World Forum on Internet of Things (WF-IoT), pp. 51–56, March 2014
16. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)
17. Kachirski, O., Guha, R.: Effective intrusion detection using multiple sensors in wireless ad hoc networks. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 8 p. IEEE (2003)
18. Mishra, A., Nadkarni, K., Patcha, A.: Intrusion detection in wireless ad hoc networks. IEEE Wirel. Commun. **11**(1), 48–60 (2004)
19. Pelechrinis, K., Iliofotou, M., Krishnamurthy, S.V.: Denial of service attacks in wireless networks: the case of jammers. IEEE Commun. Surv. Tutor. **13**(2), 245–257 (2011)
20. Basagni, S.: Distributed clustering for ad hoc networks. In: Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN 1999), pp. 310–315. IEEE (1999)
21. Karnik, N.M., Tripathi, A.R.: Security in the Ajanta mobile agent system. Softw. Pract. Exp. **31**(4), 301–329 (2001)
22. Sander, T., Tschudin, C.F.: Protecting mobile agents against malicious hosts. In: Vigna, G. (ed.) Mobile Agents and Security. LNCS, vol. 1419, pp. 44–60. Springer, Heidelberg (1998)
23. Hohl, F.: Time limited blackbox security: protecting mobile agents from malicious hosts. In: Vigna, G. (ed.) Mobile Agents and Security. LNCS, vol. 1419, pp. 92–113. Springer, Heidelberg (1998)
24. Page, J., Zaslavsky, A., Indrawan, M.: A buddy model of security for mobile agent communities operating in pervasive scenarios. In: Proceedings of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, vol. 32, pp. 17–25. Australian Computer Society, Inc. (2004)
25. Page, J., Zaslavsky, A., Indrawan, M.: Countering security vulnerabilities using a shared security buddy model schema in mobile agent communities. In: Proceedings of the First International Workshop on Safety and Security in Multi-Agent Systems (SASEMAS 2004), pp. 85–101 (2004)
26. Zikratov, I.A., Lebedev, I.S., Gurtov, A.V.: Trust and reputation mechanisms for multi-agent robotic systems. In: Balandin, S., Andreev, S., Koucheryavy, Y. (eds.) NEW2AN/ruSMART 2014. LNCS, vol. 8638, pp. 106–120. Springer, Heidelberg (2014)
27. Hong, Y., Hu, J., Gao, L.: Tracking control for multi-agent consensus with an active leader and variable topology. Automatica **42**(7), 1177–1182 (2006)
28. Ni, W., Cheng, D.: Leader-following consensus of multi-agent systems under fixed and switching topologies. Syst. Control Lett. **59**(3), 209–217 (2010)